Trumpet Winsock Version 2.1

By Peter R. Tattam

Introduction

Thank you for using the Trumpet Winsock. It is through the kind support of our users that quality networking software has been available at affordable prices to the Internet community.

You may use the Trumpet Winsock for 30 days to evaluate its usefulness. If at the end of that time you are satisfied with the Trumpet Winsock as a product, you should register it.

See Registration Details for more information.

About this documentation

This documentation has been revised for Trumpet Winsock Version 2.1 by Rob Torok.

Originally written as a Word for Windows 6.0 document, it is available in a variety of formats (including an online Help file, trumpwsk.hlp) from our Web and FTP sites.

What is the Trumpet Winsock?

The Trumpet Winsock is a Windows Sockets 1.1 compatible TCP/IP stack that provides a standard networking layer for many Windows(tm) networking applications to use, and has itself been a major vehicle in achieving widespread use of Windows Sockets 1.1.

Capabilities of the Trumpet Winsock

The Trumpet Winsock is supported under the following configurations and associated features.

1. Dial up SLIP and PPP connections.
   - Demand load dialling.
   - A Scripting Language for automatically logging in and out of your SLIP/PPP server.
   - Dynamic IP address assignment.
2. Connection to local area network by way of a packet driver and WINPKT under enhanced mode.
   - BOOTP and RARP.
3. Ethernet and SLIP packet driver types.

See Unsupported configurations if you want to use the Trumpet Winsock with any of: ODI, NDIS, Windows for Workgroups, PKTMUX, Cabletron Network Cards, Token ring, etc...

NOTE: If you already have some kind of TCP/IP networking package installed, it is likely that the Trumpet Winsock will not run. At best, you may be able to install some massaging of your system configuration. At worst, you may have to uninstall that networking package. However, if there is already a Winsock available for your package, the Trumpet Winsock will not be required.

What's new in Version 2.1?

Modifications made to the Trumpet Winsock since Version 2.0 include a few bug-fixes and:

- Revised documentation including online help.
- Improved online registration system.
- Two new scripting commands: message and status.
- More robust packet error handling.
- Less use of global DOS memory.
- Multicast support.
- Improved utilities: Ping, Hop and Dig.

License Terms and Conditions

The following terms and conditions apply to your use of the Trumpet Winsock program. Please read these terms before continuing the registration procedure. By using a registered copy of the Trumpet Winsock program you accept the terms and conditions below. If you do not accept the terms and conditions of this license, return this package to Trumpet Software International Pty Ltd and destroy all further copies of the product.

1. TSI grants the registered user of the software ("You") a non-exclusive, non-transferable and perpetual licence to use the object code comprised in the software. You must not have installed at any one time more than the number of licensed copies of the software.
2. **You acknowledge that the software is supplied "AS IS" and that TSI not making, and that no person acting on behalf of TSI has made, any warranty or representation as to the Software's functionality, performance characteristics, merchantability or suitability for any particular purpose. All such warranties and representations are hereby excluded to the maximum extent permitted by law.**
3. To the extent permitted by law, TSI, its directors, servants, agents and licensors are excluded from all forms of direct, special, indirect, or consequential loss or damage (including loss of profits, loss of data and loss or damage that may reasonably be supposed to have been in the contemplation of the parties as of the date of this agreement as a probable result of an act or omission by TSI) arising out of or in connection with Your use of the Software, including without limitation any loss or damage caused by TSI's negligence. Where relevant legislation does not permit TSI's liability to be limited in this way then, if permitted by that legislation, TSI's liability will be limited to, at TSI's option, the replacement of the relevant goods or the supplying of relevant services again, or the payment of the cost of having the services supplied again. For the purposes of this Clause 3, TSI is or will be taken to be acting as agent or trustee on behalf of and for the benefit of its directors, servants, agents and licensors.
4. The agreement formed by TSI's acceptance of Your application for registration is governed by and will be construed in accordance with the laws of the State of Tasmania, Australia. The 1980 United Nations Convention on the International Sale of Goods does not apply to that agreement.
5. You acknowledge that all rights in the Software are owned by TSI or its licensors and shall not at any time make any statement or do anything which is inconsistent with that ownership. Without limitation, Your licence does not allow You to resupply the software to any other person or to distribute the software.
6. Your licence is limited to use for Your own internal requirements and does not permit You to use the Software to process the data of any other person or a part of a bureau service. If use for these purposes is required then You should contact TSI to make arrangements for the granting of an appropriate licence covering such uses.

Installing the Trumpet Winsock

1. Although these instructions are intended for people who are new to the Internet, it is assumed that the reader has a moderate understanding of Windows and perhaps some knowledge of DOS.
   - ie. You may need some help if you don't know: a) how to edit your autoexec.bat file, or b) the difference between Program Manager and File Manager.
2. Most users will be interested in either Using the Trumpet Winsock over SLIP/PPP if they have a dial up account or Using the Trumpet Winsock with a packet driver if they are on a LAN.
   - Since the installation instructions for these two configurations are quite different, they have been split into two sections. Please read carefully through the section that is relevant to you.
3. If you need to set up your Winsock for both these situations, read about using the Command line to define alternative ini files first.

 Installation Requirements

Installation Requirements

For basic installation of the Trumpet Winsock, you will need the following:

- About 300-400k RAM (including 50-58k of global DOS memory) under normal conditions.
- About 1000k disk space.
- About 5 minutes to install the files.
- Some details for the Winsock Setup screen.

Using the Trumpet Winsock over SLIP/PPP

This section describes how to configure the Trumpet Winsock for use over SLIP, CSLIP or PPP, collectively called dial up connections.

Have a look at the relevant sections of the Trumpet Winsock FAQ, or Trumpet Winsock Error Messages for more information on this topic.

Installing the Winsock files for SLIP/PPP

1. Create a suitable directory for your Trumpet Winsock files.
   - eg. **c:\trumpet**
2. Into this directory, copy the files: tcpman.exe, winsock.dll, hosts, services, protocol, sendreg.exe, setup.cmd, login.cmd, bye.cmd and trumpwsk.hlp.
   - See Trumpet files for a description of each of these.
   - The Trumpet Winsock requires that you **must** have the correct combination of tcpman.exe, winsock.dll. When upgrading to a new release, replace each of these files to be sure that everything is up to date.
3. Modify the path line in your autoexec.bat to contain a reference to that directory.
   - eg. path c:\dos;c:\windows;**c:\trumpet**
4. Make sure it is active by rebooting your computer or by executing autoexec.bat again.
5. Check that the path is set correctly by starting Tcpman from the File Manager.
   - To do this select File/Run and enter: **tcpman**
   - If this fails, the path is probably not set up correctly, so fix it.
6. To create a program group for your Trumpet files:
   - Select File/New from the Program Manager, and choose Program Group.
   - For the Description, enter an appropriate name, eg: **Trumpet Files**
- Leave the Group File blank (Windows will set this automatically) and select OK.
7. To create a Program Item for the Tcpman within your **Trumpet Files** program group:
   - Open the File Manager and position it so that you can see both the **Trumpet Files** group (in the Program Manager) and the contents of the **trumpet** directory. You may need to resize the File Manager to do this.
- Drag tcpman.exe from the **trumpet** directory into the **Trumpet Files** program group.

Setting up the Winsock for SLIP/PPP

To enable your Winsock to function correctly it will be necessary to enter some details in the Setup screen. If you are unclear on any of them, try to seek some help from qualified Internet support staff - it will save you a lot of time.

1. If you haven't already done so, read Installing the Winsock files for SLIP/PPP.
2. Open the Trumpet Winsock by double clicking the Tcpman icon and select Setup from the File menu to open the Setup dialog.
3. Check either Internal SLIP or Internal PPP.
   - Some details will be greyed out now. You need not try to fill them in.
4. Enter your IP address (if you've been assigned one), Name server, and Domain suffix.
   - If your provider allocates IP addresses dynamically, leave this as 0.0.0.0 for the moment.
5. If you're feeling particularly brave, you may like to tweak Demand load timeout, MTU, TCP RWIN, TCP MSS, and TCP RTO MAX.
   - These values can be tailored to suit your needs once you have things running, but you should be able to establish a connection using MTU: **576**, TCP RWIN: **2048**, and TCP MSS: **512**.
6. Set SLIP port to the number of the comms port to which you've connected your modem.
   - Eg. If you're using com2, enter: **2**
7. Set the Baud rate that you would like your modem to run at.
   - Eg. If you've got a Vfast 28.8k modem, enter: **38400**
   - Up to 115200 is supported although speeds greater than 19200 require suitable hardware.
8. Set Hardware handshake and Van Jacobson CSLIP compression as required.
   - Hardware handshake is recommended if your modem cable supports it.
   - Van Jacobson CSLIP compression may only be used with SLIP if your server supports it.
9. Select the kind of Online status detection your modem supports.
   - You will need to make sure that your modem has a default power on setting of AT&C1 for DCD detection to function.
10. By default, all dialling is done with 8 bits, no parity. This may not work for you...
    - If your server does not use 8 bits with no parity, you will need to go to Dialler/Options and select Use Control Panel settings for parity and word size.
11. When you are done, click on OK and if all goes well, the Trumpet Winsock will be initialised.
    - You should now be ready to try logging in manually.

Logging in manually

Note that the Winsock will try to dial automatically when it starts unless No automatic login from Dialler/Options is selected.

1. Select Dialler/Manual Login.
   - You will now be able to send commands (as described in your modem's manual) to your modem.
2. Confirm that your modem is ready by typing an appropriate command (or initialisation string).
   - For example, try entering: **ATZ**
   - The modem should return: **OK**
   - If it doesn't, check that your modem is on, your SLIP port is set correctly, and that your cables are fine.
3. Instruct the modem to start dialling.
   - For example: **ATDT <number>**
4. Once connected to your provider's machine, you will usually be asked to enter you username and password.
5. If and when you are given a choice, select either SLIP or PPP.
   - Your screen should start being filled with nonsense characters.
6. Press <esc> to enable SLIP/PPP.
7. If you are using SLIP and your IP address is allocated dynamically you will need to enter your IP address in the Setup.
   - PPP does this automatically.
8. Try connecting to a well known host IP address using a Winsock compliant program to see if all is well.
   - The Trumpet Ping program is suitable for this.
9. If you wish to use another terminal program to dial in to the server, don't forget to issue AT&D0, or disable DTR dropping when exiting the program, or the connection will be severed when the application closes the comms port.
10. Once you have determined your login sequence, you can set up a login script to automate the process.

Automating your login sequence

Before modifying or writing your own script:

1. Find out if your Internet provider has a preconfigured a Trumpet Winsock login script.
   - Since many Internet providers will have prepared scripts for connecting to their systems, it may be worth contacting yours to see if they have a suitable script.
2. Make sure that that you've got Logging in manually under control.
   3. Try running login.cmd without any modifications.
   - The first time it is launched you will be asked to enter phone number, username and password. You can change these later by running setup.cmd.
   - If you receive a message saying that either SLIP or PPP has been established and that the script was completed then your script may require no further work.
- If you receive "script aborted" (possibly proceeded by another error message) then you will need to read on...

Here are some steps to take if you wish to personalise your login.cmd:

1. The variable $modemsetup is the initialisation string for your modem. Some research may be required to find out what this should be.
2. The other variables ($userprompt, $passprompt, etc.) will need to be altered according to the messages you provider's machine gives and accepts when you on login.
   - Eg. If the prompt that asks you for your username is "Enter your login name:", you could change the line defining this prompt to: $userprompt = "login name:"

Using the Trumpet Winsock with a packet driver

This section describes how to configure the Trumpet Winsock for use on a local area network (eg Ethernet) by way of a packet driver.

Packet drivers can only be used reliably using WINPKT, and even then only under enhanced mode. Users of standard mode must take care to avoid system crashes.

See Unsupported configurations if you want to use the Trumpet Winsock with any of: ODI, NDIS, Windows for Workgroups, PKTMUX, Cabletron Network Cards, Token ring, etc...

If your network card was supplied without a packet driver, you may like to investigate the Crynwr packet driver collection.

Have a look at the relevant sections of the Trumpet Winsock FAQ, or Trumpet Winsock Error Messages for more information on this topic.

Installing the Winsock files for a packet driver

1. Create a suitable directory for your Trumpet Winsock files.
   - eg. **c:\trumpet**
2. Into this directory, copy the files: tcpman.exe, winsock.dll, winpkt.com, hosts, services, protocol, sendreg.exe, and trumpwsk.hlp.
   - See Trumpet files for a description of each of these.
   - The Trumpet Winsock requires that you **must** have the correct combination of tcpman.exe, winsock.dll and winpkt.com. When upgrading to a new release, replace each of these files to be sure that everything is up to date.
3. Modify the path line in your autoexec.bat to contain a reference to that directory.
   - eg. path c:\dos;c:\windows**;c:\trumpet**
4. Make sure it is active by rebooting your computer or by executing autoexec.bat again.
5. Check that the path is set correctly by starting Tcpman from the File Manager.
   - To do this, open File Manager, select File/Run, and enter: **tcpman**
   - If this fails, the path is probably not set up correctly, so fix it.
6. To create a program group for your Trumpet files:
   - Select File/New from the Program Manager, and choose Program Group.
   - For the Description, enter an appropriate name, eg: **Trumpet Files**
- Leave the Group File blank (Windows will set this automatically) and select OK.
7. To create a Program Item for the Tcpman within your **Trumpet Files** program group:
   - Open the File Manager and position it so that you can see both the **Trumpet Files** group (in the Program Manager) and the contents of the **trumpet** directory. You may need to resize the File Manager to do this.
- Drag tcpman.exe from the **trumpet** directory into the **Trumpet Files** program group.

Installing WINPKT

To install a packet driver and WINPKT, you will need lines in your network batch file that do the following:

1. Set up your packet driver (including vector, IRQ and I/O address).
2. Install WINPKT using the same vector that the packet driver was installed on.

The following examples are intended to illustrate this.

Ne2000 packet driver using WINPKT

This example installs an ne2000 packet driver on vector 0x60 using IRQ 2 and I/O address 0x300.

```
ne2000 0x60 2 0x300
WINPKT 0x60
```

Western Digital 8-bit network card using WINPKT

This example installs a Western Digital 8-bit network card on vector 0x61 using IRQ 5 and I/O address 0x320.

```
wd8003e 0x61 5 0x320
WINPKT 0x61
```

Ne2000 packet driver with Novell NetWare access using WINPKT

Specification of the -n switch of the packet driver is important. Some packet drivers don't support this switch. In that case, you may be forced to use ODI instead. An example could be the Xircom Pocket Adaptor.

```
ne2000 -n 0x60 2 0x300
WINPKT 0x60
pdipx
netx
path c:\dos;c:\network\win31
f:
login
```

Setting up the Winsock for a packet driver

To enable your Winsock to function correctly it will be necessary to enter some details in the Setup screen. If you are unclear on any of them, try to seek some help from qualified Internet support staff - it will save you a lot of time.

The first time you run Tcpman, the Setup screen will be opened automatically. Select File/Setup to open it manually.

1. If you haven't already done so, read Installing the Winsock files for a packet driver and Installing WINPKT.
2. Make sure that Internal SLIP and Internal PPP are **not** checked.
   - Some details will be greyed out now. You need not try to fill them in.
3. Enter your IP address, Netmask, Default gateway, Name server, and Domain suffix.
   - If you are on a network that uses BOOTP or RARP to determine its IP address, then enter **BOOTP** or **RARP** respectively.
4. Set the Packet vector to the vector that your packet driver was installed under without the leading "0x".
   - Eg. If you used vector 0x60, enter: **60**
   - Alternatively, enter **00** to instruct the Winsock to search for the packet driver.
5. If you're feeling brave, you may like to tweak Demand load timeout, MTU, TCP RWIN, TCP MSS, and TCP RTO MAX.
   - These values can be tailored to suit your needs. If you are on an Ethernet, then we suggest MTU: **1500**, TCP RWIN: **4096**, and TCP MSS: **1024**.
6. When you are done, click on OK and if all goes well, the Trumpet Winsock will be initialised.
   - You are now ready to start using the Winsock.

Scripting Language

This section describes the Trumpet Winsock scripting language as well as providing examples of its usage.

See Automating your login sequence for some tips on how to write dialler scripts quickly.

Have a look at the relevant sections of the Trumpet Winsock FAQ, or Trumpet Winsock Error Messages if you're having difficulties with a script.

Basic commands

Each of the following commands is a scripting primitive and will return success or failure depending on whether the command succeeded or not. Commands always return success unless otherwise specified.

Parameters enclosed in < > characters need to be replaced by a value.

For example, the parameter <timeout> corresponds to a length of time in seconds and should be replaced an integer value or integer variable. Commands with a <timeout> parameter will return failure if they have been unable to succeed within <timeout> seconds, unless otherwise specified.

Parameters enclosed in ( ) characters are compulsory and those enclosed within [ ] are optional. All parameters are compulsory unless otherwise stated. The | character represents logical OR.


abort

address <timeout>

bootp

display <string>

echo [on | off]

exec <program command>

exit

expect <timeout> <target string>

input <timeout> <target string>

load <variable> [<string variable> [<ini file>]]

message <string>

online

output <string>

outputecho [timeout] <string>

password [<prompt string>]

query <string variable> [<prompt string>]

quit

read <timeout> <string variable>

readx <timeout> <string variable>

save <variable> [<string variable> [<ini file>]]

set (dtr | rts) [on | off]

sleep <timeout>

status <string>

trace [on | off]

username [<prompt string>]

wait <timeout> [dsr | cts | dcd | rlsd]

abort

Aborts the script. This command always returns failure.

See exit.

address <timeout>

Parses the current stream for an IP address, and copies it into the Winsock's primary IP address. The address change will not be effective until the script finishes.

Note that this will find the first IP address in the stream. If the current stream contains alternative IP addresses (annex IP address, etc) you will first need to check the input stream with the input command.

bootp

Attempts a BOOTP when the (login) script has finished executing. This will modify any TCP/IP setup parameters that are specified in the returned BOOTP response.

display <string>

Displays <string> in the tcpman window.

echo [on | off]

Displays any characters received while dialling in the tcpman window. The default is on.

exec <program command>

Executes the Windows command specified. This is useful for starting another program from the script.

It is important to quote literal strings by quotes if they contain the “\”(reverse slash) character. The reverse slash has a special meaning.

exit

Stops the currently executing script but still returns success.

See abort.

expect <timeout> <target string>

Checks that the current input stream contains the target string. It must match exactly, and will not search the stream. If the command timed out or the string received does not match, the command returns failure.

input <timeout> <target string>

Scans the current input stream for <target string>.

load <variable> [<string variable> [<ini file>]]

Returns <variable> which may be integer or string from <ini file>.

Where <string variable> is the section name within the ini file and defaults to “default vars”, and <ini file> is the name of the Winsock ini file and defaults to “trumpwsk.ini”.

Note that the file extension must be specified with the file name.

See save.

message <string>

Presents a dialog box containing <string>, and pauses until OK is pressed.

online

Instructs the Winsock to go online immediately. Starts processing SLIP packets, and adjusts the state of the comms port to normal SLIP conditions.

output <string>

Sends <string> to the output stream on the comms port. Any characters received during this process will be echoed to the tcpman window if the echo flag is on.

See outputecho, echo.

outputecho [timeout] <string>

Sends <string> to the output stream on the comms port. As each character is output, the script processor waits for the corresponding character to be echoed. If the received character does not match, the command will abort and a failure is returned.

If a timeout is not provided, a default of 10 seconds is used. The timeout applies to the whole command.

In addition, the carriage-return is handled specially by waiting for a linefeed to be read after the carriage return is read. Both the sequences <CR><LF> and <CR><CR><LF> are acceptable.

See output.

password [<prompt string>]

Brings up a message box that allows $password (one of the predefined variables) to be entered or modified. If the user cancels the message box, failure is returned.

Any characters typed will be echoed with a "*" character, and encrypted if saved to the .ini file.

See query, username.

query <string variable> [<prompt string>]

Brings up a message box for the <string variable> to be entered or modified. An optional prompt may be provided. If the user cancels the message box, return failure.

See password, username.

quit

Posts a quit message to the Winsock tcpman. This will start a normal quit process.

read <timeout> <string variable>

Reads a whole line (terminated by a line feed character) from the comms input stream into <string variable>. Any carriage returns are stripped.

See readx.

readx <timeout> <string variable>

Reads a whole line (terminated by a carriage return) from the comms input stream into <string variable>.

See read.

save <variable> [<string variable> [<ini file>]]

Saves <variable> which may be (integer or real) to <ini file>.

Where <string variable> is the section name within the ini file and defaults to “default vars”, and <ini file> is the name of the Winsock ini file and defaults to “trumpwsk.ini”.

Note that the file extension must be specified with the file name.

See load.

set (dtr | rts) [on | off]
Toggles the DTR or RTS modem control.

sleep <timeout>

Pauses for <timeout> seconds.

Always returns success.

status <string>

Puts <string> into the Dialler Status dialog box, replacing any previously entered text.

The Dialler Status box, which is initially left blank, is automatically displayed when any script is executed. The script aborts if Cancel is pressed.

trace [on | off]

Toggles the tracing of scripting commands. This provides a description of each line as it executes.

username [<prompt string>]

Brings up a message box that allows $username (one of the predefined variables) to be entered or modified. If the user cancels the message box, failure is returned.

See query, password.

wait <timeout> [dsr | cts | dcd | rlsd]

Pauses until the given modem signal changes state or timeout occurs.

See sleep.

String formats

When a string is required for a parameter, all characters up until the end of the command line are processed as the string. Strings conform to the following format.

If you wish to use the characters \, #, $ or % inside a string, they must be quoted with “ or have a \ character in front of them. The “ (quote) character may only be formed by preceding it with a \ character.

- “ (quote)
- # (hash)
- $ (dollar)
- % (percent)
- \ (reverse slash)

“ (quote)

A quote character (“) means that all characters are to be taken as is without any special meaning until a corresponding closing quote (“) has been found. The string is not permitted to extend over more than one line.

(hash)

If a “#” symbol is found that is not inside quotes, then it means that the rest of the logical line is to be ignored. This can be used to annotate the script.

$ (dollar)

If a "$" symbol is found, it denotes a string variable. If the variable has been assigned a value when the command is executed, it's value will be placed in the string. If not, an error message is displayed, and the script continues.

% (percent)

If a "%" symbol is found, it denotes an integer variable. If the variable has been assigned a value when the command is executed, its value will be placed in the string. If not, an error message is displayed, and the script continues. Integers are 32 bit signed integers in the range -2147483648 to 2147483647.

\ (reverse slash)

If a reverse slash "\" is found, it denotes that the character following either has a special meaning which will be placed in the string, or is to be taken literally.

Here is a list of special "\" character meanings.

\b	Back space character (control character number 8).
\c	The port number that the SLIP driver is using as a decimal number.
\e	Escape character (control character 27).
\f	Form feed character (control character 12).
\i	The current IP address.
\l	Line feed character (control character 10).
\n	An end of line sequence (control characters 13, then 10).
\p	The current password.
\r	Carriage return character (control character 13).
\t	Tab character (control character 9).
\u	The current username.
\0 to \255	Denotes a decimal number of a character to be added into the stream (eg. \0 \27 \255).

Program control commands

In addition to the basic command primitives, there are the following special control statements.

- If Statement
- While Loop
- Repeat .. Until Loop

If Statement

```
if <condition>
    <statement list>
[ else
    <statement list> ]
end
```

If the condition evaluates as true, the first statement list is executed. If the condition evaluates as false, the first statement list is ignored, and if an else clause is present, it is executed instead.

While Loop

```
while <condition>
    <statement list>
end
```

While the condition evaluates as true, the statement list is executed.

Repeat .. Until Loop

```
repeat
    <statement list>
until <condition>
```

The statement list is repeatedly executed until the condition evaluates as true.

Assignments and expressions

Assignments are used to store new values into variables. They take the following forms:

```
<integer variable> = <integer expression>
```

Eg. %attempts = %attempts + 1

```
<string variable> = <string expression>
```

Eg. $name = $first + $last

The variable part may be a string variable (either unspecified or starting with a $), or an integer variable (starting with a %). Variables aren't case sensitive.

String expressions may be composed of the following operands:

quoted strings	eg. "Fred", "OK"
string variables	eg. $name, $response

String expressions may use the following operators: ( ) + concatenation

Integer expressions may be composed of the following operands:

numbers	eg. 1, 43, 7373
integer variables	eg. %I, %count

Integer expressions use the following operators in order of priority:

( )	Brackets.
*** / %**	Multiplication, division and modulo division.
+, -	Addition and subtraction.

Also, integer operands may be formed by the comparison of strings.

eg. $A = "FRED" will evaluate to an integer operand of 0 or 1 depending on the value of $A.

Conditional expressions may also be formed using conditions and boolean operators. They return 0 or 1 depending on the result of the condition.

These operators have the following priorities.

= <> < > <= >=	Equal, not equal, less than, greater than, less than or equal, greater than or equal.
!	Boolean not operator.
&	Boolean and operator.
|	Boolean or operator.

Predefined variables

The following predefined variables are available for use within scripts.

%slipport	Corresponds to your SLIP port.
%ppp	0 = SLIP, 1 = PPP
$inipath	Determines the ini file in that variables will be saved to and loaded from.
$section	Determines the section of the ini file that variables will be saved to and loaded from.
$username	Sets the username.
$password	Sets the password. Any variable with “password” in its name will be saved in an encrypted form.

String functions

The following string functions may be used as operands.

- copy ($s, %,p %l)
- len ($s)
- lower ($s)
- pos ($s1, $s2)
- upper ($s)

copy ($s, %,p %l)

Copy characters from string $s starting at character %p for %l characters and return them as an expression. Note that string character counting starts at 1 - not zero.

len ($s)
Return the length of string $s as an integer number.

lower ($s)

Convert string $s to all lower case.

pos ($s1, $s2)

Return an integer number corresponding to the position of the first character of string $s1 as found in string $s2. A zero value is returned if $s1 is not found within $s2.

upper ($s)

Convert string $s to all upper case.

Commands as operands

A scripting command can be used as part of an expression by surrounding the command with [ ] symbols.

```
Eg. [input 10 OK\n]
```

This operand would have the value 1 if the command succeeded or 0 if it failed.

Example script fragments

- Repeated dial
- Busy detect
- Multiple number
- Transmit Break

Repeated dial

The following segment of script could be used to attempt a repeated dial of a given number. Note the use of the outputecho rather than output so that any characters echoed from the command will be consumed.

```
%attempts = 0
repeat
    %attempts = %attempts + 1
    outputecho 60 atdt345772371\r
until [input 30 CONNECT\n] | %attempts = 10
```

This section of script is fine, but would take 30 seconds for the input function to timeout if a response other than CONNECT were returned. It could be refined further into the following lines

```
%attempts = 0
repeat
    %attempts = %attempts + 1
    outputecho 60 atdt345772371\r
    read 30 $result
until $result = "CONNECT" | %attempts = 10
```

This piece of script would be fine unless the modem failed to respond, in which case the script would abort. A further refinement would be the following.

```
%attempts = 0
repeat
    %attempts = %attempts + 1
    outputecho 60 atdt345772371\r
    %timeout = [read 30 $result]
    if %timeout = 0
          display "Dial up timed out."\n
    end
until $result = "CONNECT" | %attempts = 10 | %timeout
```

These portions of script are only examples to demonstrate the use of scripting. In practice, most modems do not generate simple messages after the atdt command. You will have to skip extra lines etc., to get a working script.

Busy detect

If you are having trouble connecting to a busy site, it would be good to be able to have Trumpet Winsock redial more quickly. Your script will need to detect a BUSY signal and redial if necessary.

In the following script segment, if a BUSY signal is detected within 5 seconds then it will redial. This time may need to be increased if it takes longer for a BUSY response. If it is not busy, then it waits for 30 seconds for a CONNECT. If it gets one, it exits the loop and continues on with the script, otherwise it will redial after 30 seconds.

```
%ok = 0
%busy = 0
repeat
   output "atdt"$number\n
   %busy = [INPUT 5 BUSY]
   if !%busy
      %ok = [INPUT 30 CONNECT]
   end
until %ok
```

Multiple number

Some service providers have more than one number to dial when they are busy. In this example, four numbers are pre-defined in the script. What it does is try each number in turn until one responds with a CONNECT. If it has already tried all four, then it goes back to the first one. In this case it would repeat indefinitely, so you may want to put a limit on the number of attempts. This can also be increased or decreased to the number of dial in numbers your service provider has.

```
$number1 = "232224"
$number2 = "232225"
$number3 = "232226"
$number4 = "232227"
$number = ""

%num = 0
%ok = 0
%busy = 0

repeat
   %num = %num + 1
   if %num > 4
      %num = 1
   end
   if %num = 1
      $number = $number1
   end
   if %num = 2
      $number = $number2
   end
   if %num = 3
      $number = $number3
   end
   if %num = 4
      $number = $number4
   end
   output "atdt"$number\13
   %busy = [input 5 BUSY]
   if !%busy
      %ok = [input 30 CONNECT]
   end
until %ok
```

Transmit Break

The only way to send a break signal is to have the modem do it. This fragment switches the modem to command mode, transmits a break, and then returns to data mode.

```
output "+++"
input 10 OK\n
output "at\b o"\13
```

Menu Commands

This section describes the function of each menu command, and the options presented in each dialog box.

File

- Setup
- Firewall setup
- PPP options
- Exit

Setup

Opens the Tcpman Setup dialog box.

For some guidance in how to set these see either Setting up the Winsock for SLIP/PPP or Setting up the Winsock for a packet driver.

- IP address
- Netmask
- Default gateway
- Name server
- Time server
- Domain suffix
- Packet vector
- MTU
- TCP RWIN
- TCP MSS
- TCP RTO MAX
- Demand load timeout
- Internal SLIP
- Internal PPP
- SLIP port
- Baud rate
- Hardware handshake
- Van Jacobson CSLIP compression
- Online status detection
- None
- DCD (RLSD) check
- DSR check

IP address

Determines the IP address of your computer.

To exist on the Internet your computer must have an IP address. This address will be allocated permanently or dynamically by your service provider.

If your service provider allocates IP addresses dynamically, this will usually be set automatically if you are using PPP. If you are using SLIP, the IP address will need to be entered either manually or by a dialler script.

Alternatively, the Winsock may be able to use either BOOTP or RARP to find your IP address if the services are online.

Netmask

Sets your network mask.

A netmask is used for determining whether or not a destination IP address is on the same network.

Your netmask will often correspond to the class of your IP address. For example, if you have a Class C IP address, use: **255.255.255.0**

Default gateway

Your default Internet gateway or router.

Name server

The numerical IP address of your domain name server.

A domain name server converts names into numeric IP addresses using the domain name system (DNS). You may provide more than one address by separating the addresses with spaces.

Time server

Leave blank.

This is a presently unused list of time server IP addresses.

Domain suffix

The name of the domain in which your computer lives. This is used for looking up names via the name server.

You may provide more than one domain name by separating the names with spaces.

Packet vector

Sets the vector that your packet driver was installed under. This value should be entered without the leading “0x”.

Eg. If you used vector 0x60, enter: **60**

Leaving the packet vector set to **00** instructs the Winsock to search for the packet driver.

MTU

This should usually be set to the MTU of your network. For Ethernet, 1500 is the maximum, and is recommended. The unofficial MTU for SLIP servers is 1006.

The Maximum Transmission Unit (MTU) of a network is the greatest amount of data that can be transferred in one physical frame on that network.

Fragmentation will occur if a packet is sent across a network that has an MTU smaller than the packet's frame length. This will lead to lower performance as fragments need to be reassembled.

Since it is virtually impossible to determine the minimum MTU of the intervening networks, trial and error is usually the best way to find out.

TCP RWIN

The TCP Receive Window (RWIN) determines how much data the receiving computer is prepared to receive.

Usually 3 to 4 times the size of TCP MSS (an exact multiple if possible).

An RWIN set too large will result in greater loss of data if a packet is lost or damaged.

An RWIN set too small (ie 1*MSS) will be very slow, as each packet will have to be acknowledged before the next packet is sent. An RWIN of 2*MSS causes problems with silly window avoidance.

TCP MSS

The TCP Maximum Segment Size (MSS) defines the largest segment of TCP data that the Winsock is prepared to receive on that connection. When a connection is established, the two ends agree to use the minimum of each end's advertised value.

Because headers are typically 40 bits, your MSS should always be less at least 40 less than the MTU.

If the MSS is too low, the data/header ratio will be low.

Alternatively, a large MSS will lead to large IP datagrams. These will fragment when travelling across networks with small MTUcard83s, thereby decreasing performance. For this reason, performance may sometimes be improved by dropping the value of your MTU.

TCP RTO MAX

The TCP Retransmission Time Out Maximum (RTO MAX) sets an upper limit on the retransmission timeout.

It may be useful to lower this to perhaps 10 seconds when using interactive programs such Chat or Telnet software over a connection with a higher than average packet loss.

Demand load timeout

The number of seconds Tcpman stays loaded after all Winsock applications have finished with it.

This only applies if the Winsock has been "demand loaded". That is, the winsock.dll has been loaded before the Tcpman. This is normally how another application will launch the Winsock.

If the Winsock hasn't been demand loaded, use the SLIP inactivity timeout.

Internal SLIP

Enables internal SLIP.

Internal PPP

Enables internal PPP.

SLIP port

Sets the number of the comms port your modem is on.

Eg. If you're using com2, set this to: **2**

Baud rate
Determines the speed of the connection.

Hardware handshake

Enables RTS/CTS handshaking.

May require the AT&K3 modem command to function properly.

Van Jacobson CSLIP compression

Enables CSLIP.

Online status detection

Needed for dialler auto login / auto logout enabling.

None
No online status detection.

DCD (RLSD) check

May require AT&C1 modem command to function.

DSR check

May require AT&S1 modem command to function.

Firewall setup

The Trumpet Winsock may be used with a Socks 4.2 compatible firewall.

Enable Firewall		Enables firewall setup.
Firewall Host:	**IP address**	This will be the IP address of the firewall that is visible to the local network.
	Port	The port that the firewall will be using to communicate to the Winsock with.
Firewall User ID		Your user ID string.
Local IP addresses:	**Net**	The address of a network that will be considered local.
	Mask	The netmask of the corresponding network.

PPP options

Password Authentication Protocol (PAP) may be used if your server supports it.

Use Password Authentication Protocol (PAP)	Enables PAP.
Username	PAP username.
Password	PAP password.

Exit

Quits the TCP manager, forcing the Winsock to be unloaded.

Edit

Copy

Clear

Copy

Copies selected text on Tcpman window to the clipboard.

Clear
Clears the Tcpman window.

Special

- Info
- Kill Socket
- Register
- License

Info

Displays a list of active sockets.

Kill Socket

Allows any socket to be killed; use with caution.

Register

You should only register your copy of the Winsock if you accept the License Terms and Conditions. For this reason, they are presented before entering the Registration screen.

If you accept them, the registration screen is opened and you will be able to enter your Registration name and password to register your copy of the Trumpet Winsock. Your password is generated as a function of your registration name, so you don't get to choose it.

See Registration Details for more information on prices and distribution and details of how to register.

Note that if you've registered an earlier version of the Winsock you will need to update your password. To do this, leave old registration name and password unaltered then open the Registration via Internet screen and select Update password to 2.1.

 Registration via Internet

Registration via Internet

This feature is for sending encrypted credit card details to Trumpet Software International via the Internet. This will only work if your Winsock is functioning and your network is visible to ours.

The Winsock will only permit you to send one registration request so make sure you get your details correct. **Do not use this feature as a test.**

To send a registration request, just fill out the details carefully and click on Send registration request.

If the transfer is taking too long, it may be necessary to click on the Abort transfer button and try again later.

You will receive a password once we have received an authorisation to debit your credit card. You can select Query registration process to see how your order is progressing.

When we have processed the transaction, the automatic registration process will update your Winsock with the password. You are then fully registered. You will also receive notification via email at the time of authentication.

If this method of registration is unsuccessful, you may need to find another way to register. See How to register from Registration Details for the alternatives.

License

Displays the License Terms and Conditions.

Trace

The following trace options provide diagnostic information about the Winsock.

Use with care since some applications may crash when the traces are active. Should a program crash with stack overflow (Error 202), the Winsock may remain loaded in memory even though Tcpman has exited. It is advisable to restart windows if this happens and possibly even reboot your machine.

Also, timing measurements of the Winsock throughput will be severely affected by the trace options.

- TCP
- UDP
- IP
- ARP
- RARP
- Ethernet
- Extra detail
- Socket calls
- DNS
- Messages
- Comms
- PPP

TCP

Provides a dump of each TCP packet including:

```
time, source port->destn port, seq number, [ack number], [flags],
wind number, [options], [data size]
```

UDP

Provides a dump of each UDP packet:

```
UDP src -> dst size
```

IP

Enables IP tracing.

ARP

Enables ARP tracing.

RARP

Enables RARP tracing.

Ethernet

Adds Ethernet headers to IP, ARP and RARP traces.

Extra detail

Adds some extra detail to TCP, UDP and IP traces.

For TCP and UDP it provides a dump of the data. For UDP it also dumps broadcasts.

Socket calls

Traces each Winsock call. The subroutine parameters are displayed as well.

DNS

Traces Domain Name Server operations.

Use with care, stack overflows can be frequent.

Messages

Traces Async Socket messages.

Comms

Traces serial port communications.

PPP

Traces PPP negotiations.

Dialler

Contains functions relevant to dial up SLIP/PPP connections.

- Login
- Bye
- Other
- Manual login
- Edit scripts
- Options
- Other scripts

Login

Invokes the login.cmd dialler script.

Bye

Invokes the bye.cmd dialler script.

Other

Invokes any script. A file selection dialog of *.cmd will be displayed.

Manual login

Invokes the dialler manually. Use <esc> to exit from the manual dialler.

See Logging in manually.

Edit scripts

Invokes notepad, allowing you to edit a script.

See Scripting Language.

Options

Calls up the dialler options dialog box.

No automatic login	Disables automatic login.
Automatic login on startup only	Calls login.cmd on startup.
Automatic login and logout on demand	Calls login.cmd on startup and bye.cmd before exiting.
SLIP inactivity timeout	If the Winsock has been manually loaded, this determines the number of minutes Tcpman stays loaded after all Winsock applications have finished with it. A value of 0 disables the timeout. See Demand load timeout.
Automatic redial when disconnected	Redials once per minute while disconnected.
Use standard SLIP settings for parity and word size	Use the standard settings of 8 bits and no parity.
Use Control Panel settings for parity and word size	This will be necessary if your dial up server does not accept 8 bits and no parity when dialling in.

Other scripts

At the bottom of the Dialler menu is a list of available scripts other than login.cmd and bye.cmd. This will often include setup.cmd.

Help

Opens the relevant section of the online help file.

- Contents
- Commands
- Search
- Distribution
- Using help
- About

Contents

The Contents page of the online help is the top level entry point into the documentation.

Commands

Provides a listing of the menu commands.

Search

The Search index of the Trumpet Winsock Help is useful for finding references to specific topics in the documentation.

Distribution

Calls up the section of the help file dealing with distribution.

Using help

Opens the help file for Windows Help.

About

Displays the version number and copyright of the Trumpet Winsock.

Trumpet Winsock Error Messages

This section describes the causes and/or solutions for some common error messages.
Have a look at our Web site for a more complete list.

- Tcpman error messages
- Dialler error messages
- Packet driver error messages

Tcpman error messages

- Unable to start registration program
- Task <name> did not call WSACleanup
- Call to undefined dynalink
- TCPMAN not found

Unable to start registration program

To use the automatic registration, it will be necessary to have the file sendreg.exe in the working directory.

Task <name> did not call WSACleanup

This message is generated when an application exits without cleaning up after itself.

Call to undefined dynalink

A call has been made to a nonexistent routine in a Windows library. This could perhaps be caused by mixed Winsock versions.

TCPMAN not found

Make sure that all the Winsock files are together in one directory which is referred to in the path line of your autoexec.bat file. See either Installing the Winsock files for a packet driver or Installing the Winsock files for SLIP/PPP for details on how to do this.
Be careful not to have spare copies of winsock.dll or tcpman.exe in directories also mentioned in the path.

Dialler error messages

- script aborted
- PPP Frame check error
- Comm error [OVERRUN]
- Unable to load TCP
- Unable to perform bootp

script aborted

A “script aborted” message will be given if the script was aborted before completion, for whatever reason. For example, the script will abort if an input command times out.

If a script has aborted, it is likely that the SLIP/PPP connection hasn’t been established correctly. For that reason, no other applications will work if a script is aborted.

Ensure that your script is suitably debugged and any error conditions have been accounted for by the script.

See Automating your login sequence for instructions on how to do this.

PPP Frame check error

To check if there are any errors when a PPP packet is sent, a Frame Check Sequence (FCS) is performed. This is calculated by the sender by adding together the binary values of all the bytes in the packet and is sent as part of the packet. The receiver makes the same calculation and if the two values aren't the same, a frame error occurs.

PPP frame check errors generally occur when transferring a large amount of data over your PPP connection. They are often caused by Comm error [OVERRUN] errors.

Comm error [OVERRUN]

When accessing local disk(s), Windows halts any IO to the comm ports for a short period of time. This in turn causes the overruns, when the Winsock is unable to access the comm port. This is unfortunately a glitch in the internals of Windows, and cannot be avoided. Overrun errors can also be caused for other reasons...

When using a fast connection with a serial card that only supports lower speeds, the Winsock is unable to send data at the correct speed to the card. This also results in overruns, and the only way to fix this is to either lower the baud rate to a rate acceptable to your hardware, or upgrade your card to one with a faster UART (16550 or faster).

It should also be noted that some video drivers built around the s3 video chip hold CPU cycles and therefore cause overruns as well. s3 have apparently released driver updates that cure the problem. Several OEM vendors using the s3 based design have also followed suit.

Unable to load TCP

If you've got BOOTP set, replace your IP address with 0.0.0.0 before dialling again.

Unable to perform bootp

No reply received to sent BOOTP requests. Either no BOOTP server available or connection not properly established.

Packet driver error messages

- Packet driver must be class 1 (Blue book) or class 6 (SLIP)
- Unable to load TCP
- Unable to bind protocol 0806/0800 another TCP stack is using the packet driver
- Unable to locate WINPKT or PKTDRV virtual packet driver
- Unable to allocate network buffers critical error
- Network buffers low not critical but unadvisable
- Unable to load network buffers / network buffers low
- enablecommnotification failed - winsock will function more slowly

Packet driver must be class 1 (Blue book) or class 6 (SLIP)

This indicates that either WINPKT hasn't been loaded properly or the wrong kind of packet driver is being used. For example, token ring drivers do not work.

Unable to load TCP

Check that the packet driver loaded properly; you can do this by observing the screen as the .bat file loading the packet driver executes. This will also tell you if WINPKT managed to find it. WINPKT will also tell you if the correct vector was chosen by tcpman.

Unable to bind protocol 0806/0800 another TCP stack is using the packet driver

Remove it.

Unable to locate WINPKT or PKTDRV virtual packet driver

Check that the packet driver setting under File/Setup is correct.

See Packet driver must be class 1 (Blue book) or class 6 (SLIP).

Unable to allocate network buffers critical error

Try to free up some special driver memory by removing windows device drivers.

See unable to load network buffers / network buffers low.

Network buffers low not critical but unadvisable

Try to free up some special driver memory by removing windows device drivers.

See unable to load network buffers / network buffers low.

Unable to load network buffers / network buffers low

It will be necessary to free up more memory for the Winsock. You can try providing more DOS memory before Windows is loaded by removing unwanted TSRs or drivers. Also there can be occasions where Windows will attempt to load the full Winsock into low memory resulting in insufficient driver memory being available. The workaround for this is to try loading the Winsock at a different time. Often this error occurs when loading automatically at start up.

enablecommnotification failed - winsock will function more slowly

The "enablecommnotification" call is a call implemented in Windows 3.1 (or later) that older comm drivers (for example turbocomm) don't recognise.

The solution is to get either a new comm driver or an update of your old one. There are apparently plenty available (cybercom, winfaxpro, etc.).

At worst, the original Windows 3.1 comm driver should do the job.

Trumpet Winsock FAQ

Questions regarding tcpman, problems with menu commands, and installation difficulties.
If you can't find the answers you're looking for in either this manual or our Web site then please contact us.

See also Trumpet Winsock Error Messages.

Tcpman FAQ

Does Trumpet Winsock work with Windows 95?

Trumpet Winsock will work with Windows 95, however you may not be able to run some or all of your 32 bit TCP/IP applications.

You need to remove any other TCP/IP files, especially the Win95 winsock.dll, and replace them with the Trumpet ones.

There is an option in Tcpman properties where you can specify a startup directory. This needs to be set to the directory where your winsock.dll file is.

While this works for some people, we have no experience with Windows 95 and so can't help you with any problems that might arise from using Trumpet Winsock with Windows 95.

Trumpet Winsock under Windows NT?

Trumpet isn't supported under Windows NT. However, some people have had moderate success by replacing the standard winsock.dll with ours.

I'm not sure exactly what new problems this creates, however there are apparently side effects. Do so at your own risk!!

I’m using ODIPKT and I cannot get any response

You are probably accessing the wrong protocol. If you have the ARP trace on, you will possibly get "ARP timed out" messages. The first parameter of ODIPKT selects the correct protocol. Try adjusting this.

Dialler FAQ

Here are the answers to some commonly asked questions regarding dial-up connections and the Scripting Language that may not be covered in Using the Trumpet Winsock over SLIP/PPP.

The connection appears to be too slow compared to Xmodem

Possibly the MTU, TCP MSS and TCP RWIN settings are not right.

Turn on the IP trace to see if fragmentation is occurring on TCP connections. If so, then reduce MSS until it stops. UDP packets will still be fragmented, but nothing can be done about that. On the trace, TCP is type 6 while UDP is type 17.

Why doesn't the Trumpet Winsock disable DTR when it unloads itself?

The Winsock is designed to be demand loaded by the application. When jumping in & out of different applications with the Winsock loading & unloading, going through a new login sequence for each invocation would be tedious.

Some input commands in the script don't work.

Check for upper case/lower case conflicts. Also check for blanks at the end of the lines. Try quoting the strings to be sure.

Glossary

This section contains various bits and pieces that didn't seem to fit anywhere else.

Command line

The inipath command line option can be used to override the inipath. This is useful if you want to use the Trumpet Winsock under a variety of situations. For example, you would require one configuration for a packet driver setup and one for a SLIP connection.

Here is an example of its usage:

```
tcpman -inipath=C:\winsock\dialup.ini
```

Note: The entire path name must be given.

The command line is accessible by high-lighting the Tcpman icon from the Program Manager and selecting File/Properties from the menu.

Crynwr packet driver collection

The Crynwr packet driver collection is a comprehensive collection of public domain packet drivers, and is available by mail, FTP, email, UUCP and modem. The drivers are distributed in three files: drivers.zip, which contains executables and documentation, drivers1.zip, which contains the first half of the .ASM files, and drivers2.zip, which contains the second half of the .ASM files.

Packet driver shims

ODI can only be used via the ODIPKT shim, and NDIS via the DIS_PKT shim.

The odipkt, dis_pkt, and ndis3pkt shims are available by anonymous FTP and have directories dedicated to each of them at:

ftp://hsdndev.harvard.edu/pub/

See Unsupported configurations for examples of how to use these.

Windows Sockets 1.1

For a full description of Windows Sockets, check out Microsoft's Winsock page:

http://www.microsoft.com/pages/developer/winsock/

WINPKT

WINPKT is a virtual packet driver interface for windows.

The program WINPKT was written by some clever people on the Internet to allow a packet driver to work correctly within Windows by making sure that packets get directed to the correct virtual machine under Windows enhanced mode. It is included with our files as winpkt.com.

Trumpet files

Here is a list of files written by Trumpet Software International Pty Ltd that included with the Trumpet Winsock package.

- winsock.dll
- tcpman.exe
- trumpwsk.ini
- sendreg.exe
- hosts
- services
- protocol
- trumpwsk.hlp
- login.cmd
- setup.cmd
- bye.cmd
- tcpmeter.exe
- trumphop.exe
- trumpdig.exe
- trumping.exe

winsock.dll

The guts of the TCP/IP stack. This section does all the work, but takes its leads from tcpman.exe

tcpman.exe

The Tcpman is the part of the Trumpet Winsock that controls how the winsock.dll communicates with the Internet.

It makes use of the trumpwsk.ini file to store its settings.

trumpwsk.ini

This file normally resides in the Winsock directory rather than the windows directory since this facilitates setting up the Winsock in a networked environment.

See Command line if you wish to use more than one .ini file.

Although editing the .ini file settings directly is not advised, experienced users may find it helpful to know about the following parameters:

ip_buffers	The number of buffers for internet frames.
pkt_buffers	The number of buffers specifically for the packet driver.
slip_logging	Takes a log of a dial up SLIP/PPP session, and saves it as usage.log.
font	Sets the font of the Tcpman display.
slip_rcvbuf	Size of the outgoing comms ring buffer.
slip_sndbuf	Size of the incoming comms ring buffer.

Some network parameters can be overridden by environment variables. They have the same names as the saved parameters in trumpwsk.ini.

sendreg.exe

The automatic registration program.

See Registration via Internet for a description of how to use it.

hosts
List of host names.

services
List of Internet services.

protocol

List of Internet protocols.

trumpwsk.hlp

The Trumpet Winsock Help file was created from a Word for Windows document using Roger Hadgraft's very helpful macro, Doctor Help. Doctor Help translates any reasonable Word file into a Windows Help file (with the aid of the Help Compiler of course).

For more information regarding Doctor Help, contact roger.hadgraft@eng.monash.edu.au.

login.cmd

A sample login script.

Since everyone's login sequence will be different, it will probably be necessary for you to write or modify a script.

See Automating your login sequence for some tips on how to do this quickly.

setup.cmd

A short script that allows you to update your login phone number, username and password.
This information is saved in your trumpwsk.ini file.

bye.cmd

A sample bye script.

tcpmeter.exe

Shows the network bandwidth. Incoming traffic is displayed across the top of the screen in green, and outgoing is across the bottom in red.

Many thanks to Stephen Davies for providing the icon for Tcpmeter.

trumphop.exe

Provides a list of the gateways that are traversed by a packet on its way to a destination address.

trumpdig.exe

Provides information on a domain or address.

trumping.exe

Trumpet Ping is a diagnostic program that allows you to send test packets to a known host on the Internet. It is useful for checking your TCP/IP connection.

Definitions

- BOOTP
- CSLIP
- Firewall
- Packet driver
- PPP
- RARP
- SLIP
- Socks 4.2
- Vector
- Virtual machine

BOOTP

Bootstrap Protocol (BOOTP) is an alternative to RARP that allows a machine to obtain its IP address, gateway address, and Netmask using UDP (User Datagram Protocol) and IP.
Therefore, BOOTP can be used over a dial up SLIP/PPP connection if required.

CSLIP

Van Jacobson Compressed SLIP (CSLIP) is a more efficient version of SLIP. Its main difference is that rather than attaching complete headers to each packet, CSLIP sends the differences between headers.

Header compression is optional in PPP.

Firewall

A firewall is broadly defined as anything that protects one network from another.

Use of a firewall and a program such as Socks 4.2 can enable a computer to access the full resources of the Internet without actually existing on the Internet. It does this by having the public side of the firewall communicate to the Internet on behalf of any machines on the Local Area Network (LAN). Machines on the LAN speak to the local side of the firewall using TCP and know nothing about the full Internet. Socks bridges the gap between the two sides of the firewall.

Here are some reasons why a LAN might use a firewall to connect to the Internet:

- Without a firewall your machine's IP address (and name) can be seen by anybody else on the Internet. This information can be hidden by a firewall, since only the public side of the firewall is visible.
- Firewalls can free up space for IP addresses, as the public side of the firewall can represent the entire LAN with one IP address. As long as the machines on the LAN are only connected to the Internet through the firewall they can use any address. There are many "official non-Internet addresses" reserved for this purpose.
- The services that will be made available to users on the LAN can be regulated, as a program like Socks can choose what protocols to support.

Packet driver

A packet driver is a small piece of software which sits in between your network card and your TCP program. This provides a standard interface which many programs can use in a similar manner to BIOS calls using software interrupts.

Why is it called a packet driver? This is because modern networks send information using packets of information rather than sending information one byte or character at a time. For example, Ethernet sends information in frames of up to 1514 bytes long. The reason for sending things in packets is that information can be transmitted much more efficiently in packets.

Packet drivers are only allowed to have a software interrupt vector in the range 0x60 to 0x7F. Normally, you will pick 0x60 as the default place to install your packet driver, but certain machine configurations may make that vector unavailable. Just choose one that is free - the packet driver should tell you if you can use it or not.

Public domain packet drivers are available from the Crynwr packet driver collection.

PPP

Point to Point Protocol (PPP) is a more complicated version of SLIP that includes error detection, and optional compression.

RARP

Reverse Address Resolution Protocol (RARP) is a low level Internet protocol used by diskless machines to obtain an IP address from a server over a physical network.
An alternative to this scheme is BOOTP.

SLIP

Serial Line Internet Protocol (SLIP) is a simple protocol which allows an Async serial connection to send Internet Protocol (IP).

See PPP.

Socks 4.2

Socks is a program that permits communication across a firewall. For more information regarding Socks, see:

ftp://ftp.nec.com/pub/security/socks.cstc/

Vector

The 80x86 family of processors allows programs to communicate with the operating system through what is called a "software interrupt", which always has a number in the range 0 to 255. This is termed a "vector" and is the one of the key mechanisms to pass control to the MS-DOS operating system. Usually the vectors are expressed in hexadecimal, with the range 0x00 to 0xFF. The 0x in front of the number means that we are using hexadecimal numbers instead of decimal numbers. They may also be expressed in the notation 00H to FFH, or $00 to $FF. If you are dealing with packet drivers, hexadecimal notation is much more common, but occasionally they are expressed in decimal. Examples of software interrupts in use on PCs are 0x10 for the video BIOS, or 0x21 for calls to DOS.

Virtual machine

A virtual machine can be either the entire Windows session, or any DOS session active within Windows. Refer to the Windows system documentation for more details.

Unsupported configurations

Although we only support use of the existing Capabilities of the Trumpet Winsock, other configurations may be possible with some tweaking. The following examples demonstrate this.

Choose the one which suits you the best and modify it to your requirements. Of prime importance is the need to make sure your network card IRQ, I/O address settings and shared memory addresses don't conflict with other cards in your computer.

See the Trumpet Winsock FAQ.

- Ne2000 packet driver with Novell NetWare access using PKTMUX
- ODI setup with NetWare access
- Windows for Workgroups 3.11 Setup
- Cabletron Network Cards
- C/SLIPPER with PKTMUX
- ODI with VLM
- Token ring

Ne2000 packet driver with Novell NetWare access using PKTMUX

In this example, PKTMUX (which must be version 1.2c or later) is being used instead of WINPKT.

```
ne2000 -n 0x60 2 0x300
pktmux 4
pktdrv
pktdrv
pktdrv
pktdrv
pdipx
netx
path c:\dos;c:\network\win31
f:
login
```

ODI setup with NetWare access

You will need ODIPKT. The latest known release is 2.4 It is important that ODIPKT reference the correct protocol for IP access. This can be specified as the first parameter to ODIPKT (0=1st, 1=2nd and so forth).

See Packet driver shims.

Here's an example network attach batch file:

```
@echo off
cd \
lh lsl
lh \odi\ne2000
cd \net
lh ipxodi
lh odipkt
lh WINPKT 0x69
lh netx
path c:\dos;c:\net\win31
f:
echo on
login
```

Also, your net.cfg must be suitably configured. For example:

```
Link Support
    Buffers 8 1586
    MemPool 16384
Link Driver NE2000
    Port #1 300 20
    Int #1 2
    Frame Ethernet_II
    Frame Ethernet_802.3
    Protocol IPX 0 Ethernet_802.3
```

The ordering of the frame protocols is important for the default setup of ODIPKT. Also, users should be aware that there are two versions of ODIPKT, one released by FTP Software, and the other, a public domain one. This example refers to the public domain version. Also note that there are two programs with the same name of ne2000.com. One is a packet driver and is referred to in an earlier section. The one referred to in this section is actually an ODI driver and won't function as a packet driver at all.

Windows for Workgroups 3.11 Setup

Courtesy of B. Armstrong and Douglas W. Jones.

There is an automatic 3.11 installer available free of charge from our ftp server:

ftp://ftp.trumpet.com.au/pub/winsock/wfwsetup/twswfwg.zip/

This is provided AS IS with no warranty and is for personal use only. You will also need to pick up a copy of the DIS_PKT9 program. It currently only supports NDIS2, not NDIS3.

Cabletron Network Cards

Courtesy of Ashok Aiyar (ashok@biochemistry.bioc.crwu.edu).

“The packet driver provided by Cabletron is a little confusing as it doesn't use the same parameters as packet-drivers that use the Crynwr skeleton.

Typically the Cabletron driver is loaded as:

```
csipd_e /s:62 /h:7 /p:300
```

In this example the software interrupt is 0x62. Load WINPKT as

```
WINPKT 0x62
```

Release 11 of the Crynwr packet drivers includes a driver for Cabletron cards written by Kai Getrost using the Crynwr skeleton that uses the same parameters as the other Crynwr drivers. This driver (CTRONDNI.COM) seems to work well with E1020/1040 and E2020 Cabletron cards. Indeed I see a performance gain over the Cabletron driver. Your mileage may vary.”

C/SLIPPER with PKTMUX

Courtesy of Ashok Aiyar (ashok@biochemistry.bioc.crwu.edu).

"Although the Trumpet Winsock has built in support for C/SLIP, there are situations when in addition to Winsock applications there is a need to run packet driver applications simultaneously over a SLIP link. For such situations, PKTMUX is of utility.

Example:

```
CSLIPPER vec=65 com1 irq=04H baud=57600 ether
PKTMUX 4 65 /4  .... (support for a maximum of 4 virtual packet
drivers)
PKTDRV 60 65
```

Configure the Trumpet Winsock to use the virtual packet driver at 0x60. All other virtual packet drivers (PKTDRV) can be loaded in the DOS Windows in which they are used. They need not be loaded before entering Windows."

(Ed. Note... You may also require the use of a special comms buffer to enhance the buffering capabilities of Windows when using slipper/cslipper. It is not needed when using the internal SLIP functions of the Winsock)

ODI with VLM

Example:

```
CD \NWCLIENT
SET NWLANGUAGE=ENGLISH
LH LSL
SMC8000.COM
LH IPXODI
ODIPKT 1 96
WINPKT 0X60
VLM /MX
CD \
```

Token ring

Token ring may be available using either ODITRPKT or the ibmtoken packet driver, but is untested by TSI.

Contact Details and Online Services

The following services are intended to support our clients by providing the latest news as well as gathering feedback about our products.

Our mailboxes are for questions that are not satisfactorily addressed by this document or our online services. Before writing, please check the Trumpet Winsock FAQ lists both in this document and at our Web site.

World Wide Web

Not surprisingly, we have a Web site. Its URL is:

http://www.trumpet.com.au/

Our Web site is particularly useful as a supplement to this document by being the most up-to-date source of information. It also allows us to provide information that would be too bulky for this document if given in full.

Technical support email addresses

Before requesting technical support, please make sure you've searched through the FAQ lists both in this document and at our Web site.

To help us diagnose your problem quickly, it is important that you send any relevant details.

For example, if your problem involves a packet driver setup, please send copies of autoexec.bat, trumpwsk.ini, and any other relevant network configurations.

Choose the right mailbox for your question carefully. Misplaced email may not be answered!

reg-support@trumpet.com.au	This is where registered users may send their technical questions. As a registered user you will have priority, but **you must include your registered name** at the start of your message.
winsock-support@trumpet.com.au	Technical questions regarding the Trumpet Winsock.
newsreader-support@trumpet.com.au	Technical questions regarding the Trumpet Newsreader.

General email addresses

Choose the right mailbox for your question carefully. Misplaced email may not be answered!

info@trumpet.com.au	This is for general information about our products that isn't addressed by the other mailboxes. Typically this kind of information will be accessible from our Web site; please look there first!
sales@trumpet.com.au	Write here if you are interested in either registering your copy of one of our products, purchasing a site license, or negotiating a distribution agreement. Check our Web site first for a current price list.
registrations@trumpet.com.au	This is for **following up** registrations. Only write here if your query concerns an order in process.
trumpet-bugs@trumpet.com.au	This is for informing us of bugs in our products. We don't normally reply to these directly, so if you want answers please write to one of the technical support addresses.
suggest@trumpet.com.au	As we look for ways to improve our products, we'll refer to mail sent here. Don't expect a response though!
www-admin@trumpet.com.au	Comments and queries about our Web site.

The trumpet.* newsgroups

You can ask questions, or discuss any aspect of any Trumpet program through these Usenet newsgroups. Feedback is always welcome.

If your provider's newsroom doesn't receive these groups you can access them by pointing your newsreader at **newsroom.trumpet.com.au**.

This site maintains the following private news groups:

trumpet.announce
trumpet.bugs
trumpet.feedback
trumpet.questions

FTP and Gopher

The latest copies of all our freely accessible software and pre-releases are available by anonymous FTP from **ftp.trumpet.com.au**.

We have a Gopher site at **gopher.trumpet.com.au**.

These sites can be particularly busy at times so please be patient if you are unable to connect.

Telephone and fax numbers

Phone (Intl):	+61 02 450220
Fax (Intl):	+61 02 450210
Phone (Aus):	(002) 450220
Fax (Aus):	(002) 450210

Postal address

Trumpet Software International Pty Ltd
GPO Box 1649
Hobart
Tasmania 7001
AUSTRALIA

Registration Details

Registration of the Trumpet Winsock is encouraged since it not only funds further development of the software, but also gives you a warm fuzzy feeling deep down inside. Furthermore, it allows you to keep using the Trumpet Winsock after 30 days.

The basic registration fee for a single user version of the Trumpet Winsock is AUS$25 for Australian users and US$25 for international (non-Australian) users (valid until 31-Dec-95). See the price list for details on multi-user site licenses.

As a registered user, you will receive support (within the existing Capabilities of the Trumpet Winsock) for up to 12 months from the date of purchase. Support will include any upgrades or bug fixes released within that 12 months.

After 12 months, upgrades for previously registered users will cost 25% of the current registration fee per annum. Arrangements will also be made for conversion of smaller licenses to larger ones.

 How to register
 Price list
 Distribution Rights
 Order form

How to register

Please fill out the order form in clear print and send it along with your payment/purchase order. We accept the following methods of payment:

Automatic registration	You may email your credit card details to us using the Registration via Internet feature of the Trumpet Winsock, which located under File/Register.
Credit card	We accept Mastercard, VISA, Bankcard, American Express and Diners Club. Credit card details may be given by fax or phone. See Contact Details for our fax and phone numbers. Credit card details may only be emailed using the automatic Registration via Internet option.
Direct bank transfer	If you are interested using this method, please contact us for the necessary details.
Cheque	Cheques should be drawn in favour of **Trumpet Software International Pty Ltd** and sent to our postal address. All cheques from international users must be in US currency and drawn on US banks. If you need to draw a cheque on a non-US bank, please contact us first.
Purchase order	We accept purchase orders for amounts over US$100.

Please note that we are unable to accept US money orders.

Price list

All prices are in US dollars for international (non-Australian) users, and Australian dollars for Australian users. For example, a single license costs AUS$25 for Australian users and US$25 for everyone else.

Prices valid until 31-Dec-1995.

Single User License

1 user	$25

Multi-user Site License

1-99 users	$25 per user
100-499 users	$2500 + $10 per additional user over 100
500-999 users	$6500 + $5 per additional user over 500
1000-4999 users	$9000 + $2 per additional user over 1000
5000 - 9999 users	$17000 + $1 per additional user over 5000
10000 - 19999 users	$22000 + $0.90 per additional user over 10000
20000 - 49999 users	$31000 + $0.80 per additional user over 20000
50000 - 99999 users	$55000 + $0.70 per additional user over 50000

Multi-user Educational Site License

1-100 users	$25 per user
100+ users	$2500 per campus

A multi-user educational site license covers a single campus only. If more than one campus is to be covered, a multi-user site license costing structure with a 10% discount applies. This is restricted to students and staff using campus facilities. If dialling in off campus, students and staff must dial in through the school's services only.

A site license is a license to a specific named organisation and may not be sub-licensed or assigned to anyone else. Copies of the Winsock software obtained under a site license may only be used by employees of the licensed organisation or, in the case of an educational organisation, by employees or students of the licensed organisation. Payment under a site license is based on the number of installed users. A site license cannot be used by an Internet Service Provider to provide Internet services for its clients.

Distribution Rights

Trumpet Software International Pty Ltd reserves all rights in the Winsock program, including the rights of publication and of distribution. The Winsock program may only be made available from FTP sites which have been approved by TSI for that purpose and may only be made available from those sites in the unmodified **twsk21c.zip** file distributed by TSI. All other forms of making the program available or of distribution (and all reproductions of the Winsock program in connection with such activities) are prohibited without TSI's express written agreement.

TSI has a standard distribution agreement for use with its distributors and a standard scale of distribution charges. Please contact TSI at sales@trumpet.com.au for further information on TSI's distribution arrangements.

Order form

ORDER FORM
for Trumpet Winsock Version 2.1

Please print (or type) clearly in block capitals. Your order will not be processed if this form is not filled in correctly.

Ship to:	Bill to:
[ ]	[ ]
[ ]	[ ]
[ ]	[ ]
[ ]	[ ]
[ ]	[ ]

Please supply the following items:

[ ] Single user license.

[ ] Multi-user site licence for [_____] users.

[ ] Multi-user educational site licence for [_____] users.

Total cost (See Price list):

[_____]

Do you require a disk with the latest version of Trumpet Winsock?

[ ] 3.5" disk.

[ ] 5.25" disk.

[ ] No disk required.

How would you like to receive your password?

[ ] Sent by post.

[ ] Sent by email.

Your registration name as will appear on the program (maximum 84 characters including spaces):

[_____]

Your e-mail address (if applicable):

[_____]

Phone number: [_____]

Fax number: [_____]

Country: [_____]

Date sent:	Expected delivery date:
[_____]	[_____]